

```
1 <?php
2
3 /**
4  * @author GOLAY Brian
5  * @version 1.0 (2021/05/20)
6  * User-related functions
7  */
8
9 namespace FlightClub\sql;
10
11 use FlightClub\sql\DBConnection;
12
13 require_once 'dbConnection.php';
14
15 /**
16  * User related class that contain user related functions with database interaction
17  */
18 class UserDAO
19 {
20
21     /**
22      * Get users for making a pagination
23      *
24      * @param [type] $offset
25      * @param [type] $limit
26      * @return void
27      */
28     public static function getUsers($offset = null, $limit = null)
29     {
30         $db = DBConnection::getConnection();
31         $sql = "SELECT * FROM tbl_user";
32
33         if ($limit !== null) {
34             $sql .= " LIMIT $limit";
35         }
36         if ($offset !== null) {
37             $sql .= " OFFSET $offset";
38         }
39
40         $request = $db->prepare($sql);
41         $request->execute();
42         return $request->fetchAll();
43     }
44
45     /**
46      * Count the number of users
47      *
48      * @return int number of users
49      */
50     public static function countUsers()
51     {
52         $db = DBConnection::getConnection();
53         $sql = 'SELECT count(*) FROM tbl_user';
54         $q = $db->prepare($sql);
55         $q->execute();
56         $information = $q->fetch();
57         return $information;
58     }
59
60     /**
```

```

61      * Get all data from an user with his ID
62      *
63      * @param int $userID
64      * @return array[mixed]
65      */
66      public static function getUserByID($userID)
67      {
68          $db = DBConnection::getConnection();
69          $sql = "SELECT * FROM tbl_user WHERE Id_User = :userID";
70          $request = $db->prepare($sql);
71          $request->execute([':userID' => $userID]);
72          return $request->fetch();
73      }
74
75      /**
76      * Get all data from an user with a flight ID
77      *
78      * @param int $flightID
79      * @return array[mixed]
80      */
81      public static function getUserByFlightID($flightID)
82      {
83          $db = DBConnection::getConnection();
84          $sql = "SELECT * FROM `tbl_user-flight` join `tbl_user` on `tbl_user-
flight`.`Id_User` = `tbl_user`.`Id_User` WHERE `tbl_user-flight`.`Id_Flight` =
:flightId";
85          $request = $db->prepare($sql);
86          $request->execute([':flightId' => $flightID]);
87          return $request->fetch();
88      }
89
90      /**
91      * Get all user data from their email
92      *
93      * @param string $email
94      * @return array[mixed]
95      */
96      public static function getUserByEmail($email)
97      {
98          $db = DBConnection::getConnection();
99          $sql = "SELECT * FROM tbl_user WHERE Txt_Email = :email";
100          $request = $db->prepare($sql);
101          $request->execute([':email' => $email]);
102          return $request->fetch();
103      }
104
105      /**
106      * Add an user to the database
107      *
108      * @param string $email
109      * @param string $password
110      * @return void
111      */
112      public static function AddUsers($email, $password)
113      {
114          $db = DBConnection::getConnection();
115          $sql = "INSERT INTO `tbl_user` (`Txt_Email`, `Txt_Password_Hash`)
116          VALUES (:email, :password)";
117          $request = $db->prepare($sql);
118          $request->execute([

```

```
119         ':email' => $email,
120         ':password' => $password
121     ]);
122 }
123
124 /**
125  * Check if there is an user with the email provided in parameter
126  *
127  * @param string $email
128  * @return string
129  */
130 public static function readEmail($email)
131 {
132     $db = DBConnection::getConnection();
133     $sql = "SELECT `Txt_Email` FROM `tbl_user` WHERE `Txt_Email` = :email";
134
135     $request = $db->prepare($sql);
136     $request->execute([
137         ':email' => $email
138     ]);
139
140     $result = $request->fetch();
141     return $result;
142 }
143
144 /**
145  * Get all user data from their email
146  *
147  * @param string $email
148  * @return array[mixed]
149  */
150 public static function readUsersByEmail($email)
151 {
152     $db = DBConnection::getConnection();
153     $sql = "SELECT * FROM `tbl_user` WHERE `Txt_Email` = :email";
154
155     $request = $db->prepare($sql);
156     $request->execute([
157         ':email' => $email
158     ]);
159
160     $result = $request->fetch();
161     return $result;
162 }
163
164 /**
165  * Change the password of an user identified by his ID
166  *
167  * @param int $id
168  * @param string $password
169  * @return void
170  */
171 public static function changePassword($id, $password)
172 {
173     $db = DBConnection::getConnection();
174     $sql = "UPDATE `tbl_user` SET `Txt_Password_Hash` = :password, WHERE
`tbl_user`.`Id_User` = :id";
175
176     $query = $db->prepare($sql);
```

```
178
179     $query->execute([
180         ':id' => $id,
181         ':password' => $password
182     ]);
183 }
184 }
185
```